

CRS-1和IOS XP可操作的最佳实践

目录

[简介](#)

[先决条件](#)

[要求](#)

[使用的组件](#)

[规则](#)

[思科IOS XR概述](#)

[进程和线程](#)

[进程和线程状态](#)

[同步消息传递](#)

[受阻进程和进程状态](#)

[重要流程及其功能](#)

[内带奥](#)

[组服务流程\(GSP\)](#)

[BCDL批量内容下载程序](#)

[轻量消息\(LWM\)](#)

[恩夫蒙](#)

[CRS-1交换矩阵简介](#)

[交换矩阵平面](#)

[交换矩阵监控](#)

[控制平面概述](#)

[Catalyst 6500配置](#)

[多机箱控制平面管理](#)

[ROMMON和Monlib](#)

[升级说明](#)

[PLIM和MSC概述](#)

[PLIM超订用](#)

[配置管理](#)

[安全](#)

[LPTS](#)

[如何转发内部数据包？](#)

[带外](#)

[相关信息](#)

[简介](#)

本文档可帮助您了解以下内容：

- 进程和线程
- CRS-1交换矩阵
- 控制层面
- 罗蒙和蒙利卜
- 物理层接口模块(PLIM)和模块化服务卡(MSC)
- 配置管理
- 安全
- 带外
- 简单网络管理协议 (SNMP)

[先决条件](#)

[要求](#)

思科建议您了解Cisco IOS®XR。

[使用的组件](#)

本文档中的信息基于以下软件和硬件版本：

- Cisco IOS XR 软件
- CRS-1

本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原始（默认）配置。如果您使用的是真实网络，请确保您已经了解所有命令的潜在影响。

[规则](#)

有关文档规则的详细信息，请参阅 [Cisco 技术提示规则](#)。

[思科IOS XR概述](#)

思科IOS XR旨在扩展。内核是微内核架构，因此它只提供基本服务，如进程管理、调度、信号和计时器。所有其他服务（如文件系统、驱动程序、协议栈和应用程序）都被视为资源管理器，并在内存保护用户空间中运行。这些其他服务可在运行时添加或删除，具体取决于程序设计。微内核的占用空间仅为12 kb。微内核和底层操作系统来自QNX软件系统，称为Neutrino。QNX专门从事实时操作系统设计。微内核优先，调度程序优先。这可确保进程之间的情景交换非常快，并且最高优先级的线程始终能够在需要时访问CPU。这些是Cisco IOS XR利用的一些优势。但是，最大的好处是操作系统核心内部进程间通信的继承设计。

中微子是消息传递操作系统，消息是所有线程之间进程间通信的基本手段。当特定服务器要提供服务时，它会创建用于交换消息的通道。客户端通过直接映射到相关文件描述符来连接到服务器通道，以便利用服务。客户端与服务器之间的所有通信都采用相同的机制。这对CRS-1的超级计算机来说是巨大的优势。在标准UNIX内核上执行本地读取操作时，请考虑以下几点：

- 软件中断到内核。
- 内核调度到文件系统。
- 接收数据。

在远程情况下，请考虑以下几点：

- 软件中断到内核。
- 内核调度NFS。
- NFS调用网络组件。
- 远程派送网络组件。
- NFS称为。
- 内核调度文件系统。

本地读取和远程读取的语义不同。文件锁定和设置权限的参数和参数不同。

考虑QNX本地案例：

- 软件中断到内核。
- 内核执行消息传递到文件系统。

考虑非本地情况：

- 软件中断到内核。
- 内核进入QNET，即IPC传输机制。
- QNET进入内核。
- 内核调度文件系统。

与参数传递和文件系统参数相关的所有语义都是相同的。IPC接口上的所有设备都已分离，允许客户端和服务端完全分离。这意味着任何进程都可以在任意时间点运行。如果特定路由处理器太忙于处理请求，您可以轻松地将这些服务迁移到在DRP上运行的不同CPU。在不同CPU上运行不同服务的超级计算机分布在多个节点上，可轻松与任何其他节点通信。为了提供扩展机会，基础设施已部署到位。思科利用了这一优势并编写了附加软件，这些软件将消息传递内核的原则操作连接到CRS路由器可扩展到数千个节点，在这种情况下，节点(CPU)运行操作系统实例，无论是路由进程(RP)、分布式路由处理器(DRP)、模块化服务卡(MSC)还是A交换机处理器(SP)。

进程和线程

在Cisco IOS XR的界限内，进程是包含一个或多个线程的内存保护区域。从程序员的角度来看，线程会完成工作，每个线程都完成逻辑执行路径以执行特定任务。线程在执行流程中所需的内存属于它们在其中运行的进程，不受任何其他进程线程的保护。线程是执行单元，其执行上下文包括堆栈和寄存器。进程是共享虚拟地址空间的一组线程，虽然进程可以包含单个线程，但更常见的是包含更多线程。如果不同进程中的另一个线程尝试在进程中写入内存，则违规进程将被终止。如果进程中运行了多个线程，则该线程有权访问进程中的同一内存，因此能够覆盖另一个线程的数据。完成过程中的步骤，以保持与资源的同步，以防止同一进程中出现此线程。

线程使用名为互斥(MUTEX)的对象来确保对服务的互斥。具有MUTEX的线程是可以写入特定内存区域的线程，例如。没有MUTEX的其他线程无法。还有其他机制来确保与资源同步，这些机制包括信号、条件变量或条件、障碍和梦游。此处未讨论这些内容，但它们提供同步服务作为其职责的一部分。如果将此处讨论的原则等同于Cisco IOS，则Cisco IOS是一个运行许多线程的单个进程，其中所有线程都可访问相同的内存空间。但是，Cisco IOS会调用这些线程进程。

进程和线程状态

在Cisco IOS XR中，有提供服务的服务器和使用服务的客户端。特定进程可以有許多提供相同服务的线程。另一个进程可以有許多客户端，这些客户端可能需要在任何时间点提供特定服务。对服务器的访问并不总是可用的，如果客户端请求访问服务，它会请求访问该服务并等待服务器空闲。在这种情况下，客户端被阻止。这称为阻塞客户端服务器模型。客户端可能会被阻止，因为它等待MUTEX等资源，或者由于服务器尚未回复。

发出show process ospf命令，以检查ospf进程中线程的状态：

```
RP/0/RP1/CPU0:CWDCRS#show process ospf
      Job Id: 250
      PID: 110795
      Executable path: /disk0/hfr-rout-3.2.3/bin/ospf
      Instance #: 1
      Version ID: 00.00.0000
      Respawn: ON
      Respawn count: 1
      Max. spawns per minute: 12
      Last started: Tue Jul 18 13:10:06 2006
      Process state: Run
      Package state: Normal
      Started on config: cfg/gl/ipv4-ospf/proc/101/ord_a/routerid
      core: TEXT SHARED MEM MAIN MEM
      Max. core: 0
      Placement: ON
      startup_path: /pkg/startup/ospf.startup
      Ready: 1.591s
      Available: 5.595s
      Process cpu time: 89.051 user, 0.254 kernel, 89.305 total
JID   TID  Stack pri state      HR:MM:SS:MSEC NAME
250   1    40K  10 Receive    0:00:11:0509 ospf
250   2    40K  10 Receive    0:01:08:0937 ospf
250   3    40K  10 Receive    0:00:03:0380 ospf
250   4    40K  10 Condvar   0:00:00:0003 ospf
250   5    40K  10 Receive    0:00:05:0222 ospf
```

请注意，ospf进程的作业ID(JID)为250。在运行的路由器上，这一情况从未改变，通常在特定版本的Cisco IOS XR上。在ospf进程中，每个线程有五个线程，各自具有自己的线程ID(TID)。列出了每个线程的堆栈空间、每个线程的优先级及其状态。

同步消息传递

前面提到QNX是消息传递操作系统。它实际上是一个同步消息传递操作系统。许多操作系统问题都反映在同步消息中。不是说同步消息传递会导致任何问题，而是问题症状反映在同步消息传递中。由于它是同步的，因此CRS-1操作员可以轻松访问生命周期或状态信息，这有助于进行故障排除过程。消息传递生命周期与以下内容类似：

- 服务器创建消息通道。
- 客户端连接到服务器的通道（类似于posix open）。
- 客户端向服务器(MsgSend)发送消息，并等待回复和阻止。
- 服务器从客户端接收(MsgReceive)消息，处理该消息，并回复客户端。
- 客户端取消阻止并处理来自服务器的应答。

此阻塞客户端—服务器模型是同步消息传递。这意味着客户端发送消息和块。服务器接收消息，处理消息，回复客户端，然后解除客户端阻止。具体细节如下：

- 服务器在RECEIVE状态下等待。
- 客户端向服务器发送消息并变为BLOCKED。
- 服务器接收消息并取消阻止（如果处于接收状态）。
- 客户端将进入REPLY状态。
- 服务器将进入RUNNING状态。
- 服务器处理该消息。
- 服务器回复客户端。

- 客户端取消阻止。

发出show process命令以查看客户端和服务器的状态。

RP/0/RP1/CPU0:CWDCRS#show processes

JID	TID	Stack	pri	state	HR:MM:SS:MSEC	NAME
1	1	0K	0	Ready	320:04:04:0649	procnto-600-smp-cisco-instr
1	3	0K	10	Nanosleep	0:00:00:0043	procnto-600-smp-cisco-instr
1	5	0K	19	Receive	0:00:00:0000	procnto-600-smp-cisco-instr
1	7	0K	19	Receive	0:00:00:0000	procnto-600-smp-cisco-instr
1	8	0K	19	Receive	0:00:00:0000	procnto-600-smp-cisco-instr
1	11	0K	19	Receive	0:00:00:0000	procnto-600-smp-cisco-instr
1	12	0K	19	Receive	0:00:00:0000	procnto-600-smp-cisco-instr
1	13	0K	19	Receive	0:00:00:0000	procnto-600-smp-cisco-instr
1	14	0K	19	Receive	0:00:00:0000	procnto-600-smp-cisco-instr
1	15	0K	19	Receive	0:00:00:0000	procnto-600-smp-cisco-instr
1	16	0K	10	Receive	0:02:01:0207	procnto-600-smp-cisco-instr
1	17	0K	10	Receive	0:00:00:0015	procnto-600-smp-cisco-instr
1	21	0K	10	Receive	0:00:00:0000	procnto-600-smp-cisco-instr
1	23	0K	10	Running	0:07:34:0799	procnto-600-smp-cisco-instr
1	26	0K	10	Receive	0:00:00:0001	procnto-600-smp-cisco-instr
1	31	0K	10	Receive	0:00:00:0001	procnto-600-smp-cisco-instr
1	33	0K	10	Receive	0:00:00:0000	procnto-600-smp-cisco-instr
1	39	0K	10	Receive	0:13:36:0166	procnto-600-smp-cisco-instr
1	46	0K	10	Receive	0:06:32:0015	procnto-600-smp-cisco-instr
1	47	0K	56	Receive	0:00:00:0029	procnto-600-smp-cisco-instr
1	48	0K	10	Receive	0:00:00:0001	procnto-600-smp-cisco-instr
1	72	0K	10	Receive	0:00:00:0691	procnto-600-smp-cisco-instr
1	73	0K	10	Receive	0:00:00:0016	procnto-600-smp-cisco-instr
1	78	0K	10	Receive	0:09:18:0334	procnto-600-smp-cisco-instr
1	91	0K	10	Receive	0:09:42:0972	procnto-600-smp-cisco-instr
1	95	0K	10	Receive	0:00:00:0011	procnto-600-smp-cisco-instr
1	103	0K	10	Receive	0:00:00:0008	procnto-600-smp-cisco-instr
74	1	8K	63	Nanosleep	0:00:00:0001	wd-mbi
53	1	28K	10	Receive	0:00:08:0904	dllmgr
53	2	28K	10	Nanosleep	0:00:00:0155	dllmgr
53	3	28K	10	Receive	0:00:03:0026	dllmgr
53	4	28K	10	Receive	0:00:09:0066	dllmgr
53	5	28K	10	Receive	0:00:01:0199	dllmgr
270	1	36K	10	Receive	0:00:36:0091	qsm
270	2	36K	10	Receive	0:00:13:0533	qsm
270	5	36K	10	Receive	0:01:01:0619	qsm
270	7	36K	10	Nanosleep	0:00:22:0439	qsm
270	8	36K	10	Receive	0:00:32:0577	qsm
67	1	52K	19	Receive	0:00:35:0047	pkgfs
67	2	52K	10	Sigwaitinfo	0:00:00:0000	pkgfs
67	3	52K	19	Receive	0:00:30:0526	pkgfs
67	4	52K	10	Receive	0:00:30:0161	pkgfs
67	5	52K	10	Receive	0:00:25:0976	pkgfs
68	1	8K	10	Receive	0:00:00:0003	devc-pty
52	1	40K	16	Receive	0:00:00:0844	devc-conaux
52	2	40K	16	Sigwaitinfo	0:00:00:0000	devc-conaux
52	3	40K	16	Receive	0:00:02:0981	devc-conaux
52	4	40K	16	Sigwaitinfo	0:00:00:0000	devc-conaux
52	5	40K	21	Receive	0:00:03:0159	devc-conaux
65545	2	24K	10	Receive	0:00:00:0487	pkgfs
65546	1	12K	16	Reply	0:00:00:0008	ksh
66	1	8K	10	Sigwaitinfo	0:00:00:0005	pipe
66	3	8K	10	Receive	0:00:00:0000	pipe
66	4	8K	16	Receive	0:00:00:0059	pipe
66	5	8K	10	Receive	0:00:00:0149	pipe
66	6	8K	10	Receive	0:00:00:0136	pipe
71	1	16K	10	Receive	0:00:09:0250	shmwin_svr

```

71      2      16K  10 Receive      0:00:09:0940 shmwin_svr
61      1       8K  10 Receive      0:00:00:0006 mqueue

```

受阻进程和进程状态

发出 `show process blocked` 命令，以查看处于阻塞状态的进程。

```

RP/0/RP1/CPU0:CWDCRS#show processes blocked
  Jid      Pid Tid      Name State Blocked-on
65546     4106 1      ksh Reply 4104 devc-conaux
  105     61495 2      attachd Reply 24597 eth_server
  105     61495 3      attachd Reply 8205 mqueue
  316     65606 1      tftp_server Reply 8205 mqueue
  233     90269 2      lpts_fm Reply 90223 lpts_pa
  325     110790 1      udp_snmpd Reply 90257 udp
  253     110797 4      ospfv3 Reply 90254 raw_ip
  337     245977 2      fdiagd Reply 24597 eth_server
  337     245977 3      fdiagd Reply 8205 mqueue
65762     5996770 1      exec Reply 1 kernel
65774     6029550 1      more Reply 8203 pipe
65778     6029554 1      show_processes Reply 1 kernel
RP/0/RP1/CPU0:CWDCRS#

```

同步消息传递使您能够轻松跟踪不同线程之间进程间通信的生命周期。线程可以随时处于特定状态。阻塞状态可能是问题的症状。这并不意味着如果线程处于阻塞状态，则出现问题，因此请勿发出 `show process blocked` 命令，并向Cisco技术支持部门提交问题。阻塞的线程也非常正常。

注意上一输出。如果查看列表中的第一个线程，请注意它是ksh，其回复在devc-conaux上被阻止。客户端（本例中为ksh）向devc-conaux进程发送消息，服务器(devc-conaux)保留ksh回复，直到其回复。Ksh是控制台或AUX端口上有人使用的UNIX外壳。Ksh等待来自控制台的输入，如果没有输入，因为操作员未键入，则它会一直被阻止，直到它处理某些输入。处理后，ksh返回devc-conaux上被阻止的回复。

这是正常的，不说明问题。要点是，被阻止的线程是正常的，它取决于XR版本、您拥有的系统类型、您配置了什么以及谁执行了哪些操作来改变 `show process blocked` 命令的输出。使用 `show process blocked` 命令是开始排除操作系统类型故障的好方法。如果出现问题（例如CPU使用率较高），则使用上一命令查看是否有异常情况。

了解正常运行的路由器的正常情况。这为在排除流程生命周期故障时用作比较的基准提供了依据。

随时，线程都可能处于特定状态。下表提供状态列表：

如果状态为：	线程为：
死亡	死了。内核正在等待释放线程资源。
运行	在CPU上主动运行
就绪	未在CPU上运行，但已准备好运行
已停止	已暂停（SIGSTOP信号）
发送	正在等待服务器接收消息
接收	正在等待客户端发送消息
REPLY（应答）	正在等待服务器回复消息
堆栈	正在等待分配更多堆栈
等待页	正在等待进程管理器解决页面故障
SIGSUSPEND	等待信号

SIGWAITINFO	等待信号
纳诺斯利普	睡一段时间
互斥	正在等待获取互斥锁
CONDVAR	正在等待条件变量发出信号
加入	等待其他线程完成
INTR	正在等待中断
SEM	正在等待获取信号

重要流程及其功能

Cisco IOS XR有许多进程。这些是一些重要的，其功能在此处说明。

[WatchDog系统监控器\(WDSysmon\)](#)

这是用于检测进程挂起和内存不足情况的服务。内存不足可能是由于内存泄漏或某些其他外部情况造成的。挂起可能是许多情况的结果，如进程死锁、无限循环、内核锁定或调度错误。在任何多线程环境中，系统都可以处于死锁状态，或仅仅是死锁状态。当一个或多个线程由于资源争用而无法继续时，可能会发生死锁。例如，线程A可以向线程B发送消息，而线程B同时向线程A发送消息。两个线程彼此等待，并且可能处于发送阻塞状态，并且两个线程会永远等待。这是一个涉及两个线程的简单案例，但如果服务器负责由多个线程使用的资源在另一个线程上被阻止，则请求访问该资源的多个线程可能会被阻止，等待在服务器上发送。

死锁可能在几个线程之间发生，但会因此影响其他线程。通过良好的程序设计可避免死锁，但无论程序的设计和编写有多么伟大。有时，与特定计时相关的特定事件序列可能会导致死锁。死锁并非总是确定性的，通常很难重现。WDSysmon有许多线程，其中一个线程以Neutrino支持的最高优先级运行，63。以优先级63运行可确保线程在基于优先级的抢占式调度环境中获得CPU时间。WDSysmon与硬件监视程序功能配合工作，并监视查找挂起情况的软件进程。检测到此类情况时，WDSysmon会收集有关该情况的进一步信息，可以重新转储进程或内核、写出到系统日志、运行脚本和终止死锁进程。根据问题的严重程度，它可以启动路由处理器切换以保持系统运行。

```
RP/0/RP1/CPU0: CWD CRS# show processes wdsysmon
    Job Id: 331
      PID: 36908
    Executable path: /disk0/hfr-base-3.2.3/sbin/wdsysmon
      Instance #: 1
      Version ID: 00.00.0000
      Respawn: ON
      Respawn count: 1
    Max. spawns per minute: 12
      Last started: Tue Jul 18 13:07:36 2006
      Process state: Run
      Package state: Normal
        core: SPARSE
      Max. core: 0
      Level: 40
      Mandatory: ON
      startup_path: /pkg/startup/wdsysmon.startup
      memory limit: 10240
      Ready: 0.705s
    Process cpu time: 4988.295 user, 991.503 kernel, 5979.798 total
```

```
JID   TID  Stack pri state          HR:MM:SS:MSEC NAME
331   1    84K  19 Receive        0:00:00:0029 wdsysmon
```

```

331    2      84K  10 Receive      0:17:34:0212 wdsysmon
331    3      84K  10 Receive      0:00:00:0110 wdsysmon
331    4      84K  10 Receive      1:05:26:0803 wdsysmon
331    5      84K  19 Receive      0:00:06:0722 wdsysmon
331    6      84K  10 Receive      0:00:00:0110 wdsysmon
331    7      84K  63 Receive      0:00:00:0002 wdsysmon
331    8      84K  11 Receive      0:00:00:0305 wdsysmon
331    9      84K  20 Sem          0:00:00:0000 wdsysmon

```

进程WDSysmon有九个线程。四个以优先级10运行，另外四个以优先级11、19、20和63运行。在设计流程时，程序员会仔细考虑流程中每个线程应指定的优先级。如前所述，调度程序是基于优先级的，这意味着更高优先级的线程总是优先于较低优先级的线程。优先级63是线程可以运行的最高优先级，在本例中为线程7。线程7是观察程序线程，跟踪CPU主机的线程。它必须以比它所监视的其他线程更高的优先级运行，否则它可能根本没有机会运行，这会阻止它执行其设计要执行的步骤。

内蒂奥

在Cisco IOS中，有快速交换和进程交换的概念。快速交换使用CEF代码，并在中断时发生。进程交换使用ip_input (IP交换代码)，并且是计划进程。在较高端平台上，CEF交换在硬件中完成，ip_input在CPU上安排。Cisco IOS XR中的ip_input等效于Netio。

```

P/0/RP1/CPU0:CWDRCRS#show processes netio
      Job Id: 241
      PID: 65602
      Executable path: /disk0/hfr-base-3.2.3/sbin/netio
      Instance #: 1
      Args: d
      Version ID: 00.00.0000
      Respawn: ON
      Respawn count: 1
      Max. spawns per minute: 12
      Last started: Tue Jul 18 13:07:53 2006
      Process state: Run
      Package state: Normal
      core: DUMPFALLBACK COPY SPARSE
      Max. core: 0
      Level: 56
      Mandatory: ON
      startup_path: /pkg/startup/netio.startup
      Ready: 17.094s
      Process cpu time: 188.659 user, 5.436 kernel, 194.095 total

```

JID	TID	Stack	pri	state	HR:MM:SS:MSEC	NAME
241	1	152K	10	Receive	0:00:13:0757	netio
241	2	152K	10	Receive	0:00:10:0756	netio
241	3	152K	10	Condvar	0:00:08:0094	netio
241	4	152K	10	Receive	0:00:22:0016	netio
241	5	152K	10	Receive	0:00:00:0001	netio
241	6	152K	10	Receive	0:00:04:0920	netio
241	7	152K	10	Receive	0:00:03:0507	netio
241	8	152K	10	Receive	0:00:02:0139	netio
241	9	152K	10	Receive	0:01:44:0654	netio
241	10	152K	10	Receive	0:00:00:0310	netio
241	11	152K	10	Receive	0:00:13:0241	netio
241	12	152K	10	Receive	0:00:05:0258	netio

组服务流程(GSP)

任何一台超级计算机都需要与数千个节点进行通信，每个节点都运行自己的内核实例。在Internet中，一对多通信通过组播协议高效完成。GSP是用于CRS-1中IPC的内部组播协议。GSP提供一对多

可靠的组通信，该通信采用异步语义无连接。这允许GSP扩展到数千个节点。

```
RP/0/RP1/CPU0:CWDCRS#show processes gsp
      Job Id: 171
          PID: 65604
      Executable path: /disk0/hfr-base-3.2.3/bin/gsp
      Instance #: 1
      Version ID: 00.00.0000
      Respawn: ON
      Respawn count: 1
      Max. spawns per minute: 12
      Last started: Tue Jul 18 13:07:53 2006
      Process state: Run
      Package state: Normal
          core: TEXT SHARED MEM MAIN MEM
      Max. core: 0
      Level: 80
      Mandatory: ON
      startup_path: /pkg/startup/gsp-rp.startup
      Ready: 5.259s
      Available: 16.613s
      Process cpu time: 988.265 user, 0.792 kernel, 989.057 total

JID  TID  Stack pri state      HR:MM:SS:MSEC NAME
171  1    152K  30  Receive    0:00:51:0815 gsp
171  3    152K  10  Condvar    0:00:00:0025 gsp
171  4    152K  10  Receive    0:00:08:0594 gsp
171  5    152K  10  Condvar    0:01:33:0274 gsp
171  6    152K  10  Condvar    0:00:55:0051 gsp
171  7    152K  10  Receive    0:02:24:0894 gsp
171  8    152K  10  Receive    0:00:09:0561 gsp
171  9    152K  10  Condvar    0:02:33:0815 gsp
171  10   152K  10  Condvar    0:02:20:0794 gsp
171  11   152K  10  Condvar    0:02:27:0880 gsp
171  12   152K  30  Receive    0:00:46:0276 gsp
171  13   152K  30  Receive    0:00:45:0727 gsp
171  14   152K  30  Receive    0:00:49:0596 gsp
171  15   152K  30  Receive    0:00:38:0276 gsp
171  16   152K  10  Receive    0:00:02:0774 gsp
```

[BCDL批量内容下载程序](#)

BCDL用于向RP和MSC等各种节点可靠地组播数据。它使用GSP作为底层传输。BCDL保证消息的传递顺序。在BCDL中，有一个代理、一个生产者和一个消费者。代理是在数据组播给消费者之前与生产者通信以检索和缓冲数据的过程。生产者是生产每个人想要的数据的过程，而消费者是希望接收生产者提供的数据的过程。BCDL用于Cisco IOS XR软件升级。

[轻量消息\(LWM\)](#)

LWM是思科创建的消息传送形式，旨在在进程间相互通信的应用和Neutrino之间创建抽象层，目标是独立于操作系统和传输层。如果思科希望将操作系统供应商从QNX更改为其他人，则底层操作系统基本功能之间的抽象层有助于消除对操作系统的依赖，并有助于移植到其他操作系统。LWM提供同步保证消息传送，这与本地Neutrino消息传送类似，会导致发送方在接收方回复之前被阻止。

LWM还通过40位脉冲提供异步消息传送。异步消息异步发送，这意味着消息已排队且发送方不会阻止，但服务器不会异步接收，而是当服务器轮询下一个可用消息时。LWM以客户端/服务器的形式结构。服务器创建一个信道，该信道让它能够侦听消息，并在环路执行侦听信道上的消息接收时处于一个循环中，该信道是它刚刚创建的。当消息到达时，它会取消阻止并获取客户端标识符，这实际上与从收到的消息接收ID相同。然后，服务器执行一些处理，随后对客户端标识符执行消息应答

。

在客户端，它执行消息连接。它会被传递到它所连接的标识符，然后发送消息并被阻止。当服务器完成处理后，它会回复，客户端将被解除阻止。这几乎与中微子本地消息传递的内容相同，因此抽象层非常薄。

LWM设计时系统调用和情景交换机的数量最少，可实现高性能，是Cisco IOS XR环境中IPC的首选方法。

恩夫蒙

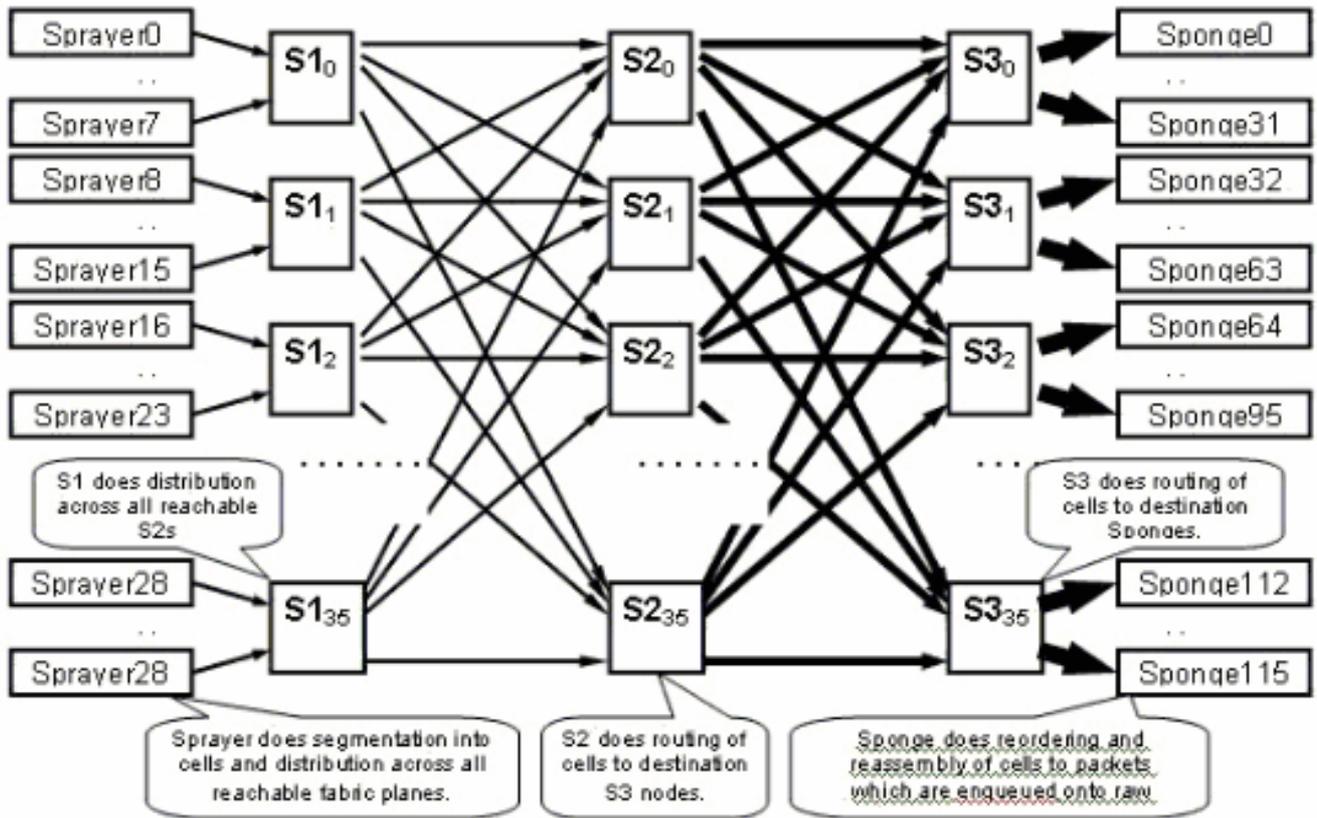
在最基本的层面，环境监控系统负责在物理参数（例如温度、电压、风扇速度等）超出操作范围时发出警告，并关闭接近硬件可能损坏的临界级别的硬件。它定期监控每个可用硬件传感器，将测量值与卡特定阈值进行比较，并根据需要发出警报以完成此任务。持续进程，在系统初始化时启动，定期轮询机箱中的所有硬件传感器（例如电压、温度和风扇速度），并将此数据提供给外部管理客户端。此外，定期过程会将传感器读数与警报阈值进行比较，并将环境警报发布到系统数据库，以便故障管理器执行后续操作。如果传感器读数超出范围，环境监控过程可能导致卡关闭。

CRS-1交换矩阵简介

- 多级交换矩阵 — 3阶段Benes拓扑
- 交换矩阵内的动态路由，以最大限度地减少拥塞
- 基于单元格：136字节单元，120字节数据负载
- 流量控制，可改善流量隔离并最大限度地减少交换矩阵中的缓冲要求
- 阶段到阶段加速交付
- 支持两种流量广播（单播和组播）
- 每播支持的流量的两个优先级（高和低）
- 支持1M交换矩阵组播组(FGID)
- 经济高效的容错：使用交换矩阵平面与1+1的N+1或N+k冗余，成本大幅增加

在单机箱模式下运行时，S1、S2和S3系列位于同一交换矩阵卡上。此卡通常也称为**S123卡**。在多机箱配置中，S2被分离，并位于交换矩阵卡机箱(FCC)上。此配置需要两个交换矩阵卡才能形成一个平面、一个S2卡和一个S13卡。每个MSC连接到八个交换矩阵平面以提供冗余，这样，如果您松开一个或多个平面，则交换矩阵仍会传递流量，尽管可通过交换矩阵的聚合流量较低。CRS对于大多数数据包大小仍可以以线速运行，只有七个平面。背压通过奇偶平面在织物上发送。不建议在奇偶平面中运行少于两个平面的系统。不支持任何少于两个平面的配置。

交换矩阵平面



上图表示一个平面。你得把那张图乘以八。这意味着LC的喷雾器(ingressq)基本连接到8个S1 (每平面1个S1)。每个交换矩阵平面中的S1连接到8个喷雾器：

- 机箱的8个顶部LC
- 8个底部LC

每个16插槽LC机箱有16个S1:顶部LC为8 (每平面1个)，底部LC为8。

在单个16插槽机箱上，S123交换矩阵卡有2个S1、2个S2和4个S3。这是交换矩阵加速计算的一部分。流量是流量可以进入的流量的两倍，可以从交换矩阵中退出。目前，每液晶还有两个纱布 (fabricq)，而1个喷雾器。这允许当多个入口LC过载出口LC时在出口LC上缓冲。出口LC能够从交换矩阵中吸收该额外带宽。

交换矩阵监控

平面可用性和连接：

```
admin show controller fabric plane all
admin show controller fabric connectivity all detail
```

检查平面是否正在接收/发送信元，并且某些错误在增加：

```
admin show controllers fabric plane all statistics
```

前一命令中的缩写：

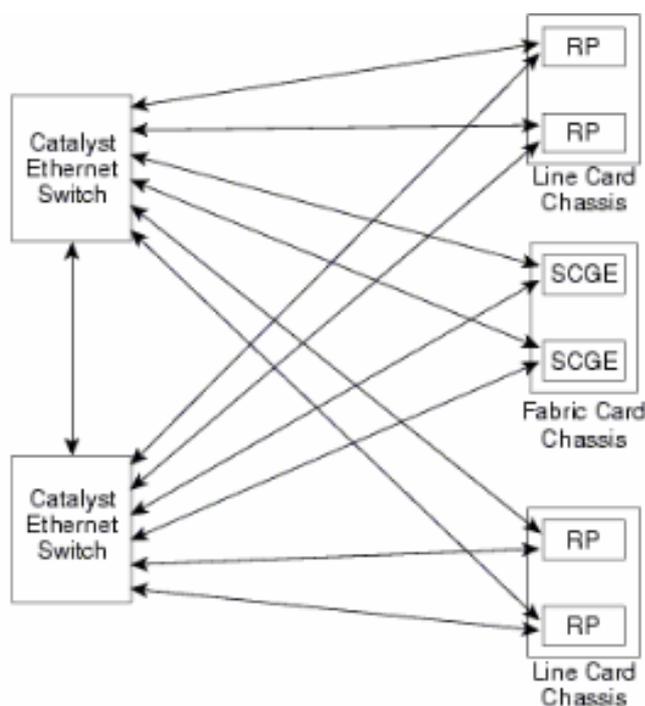
- CE — 可纠正错误
- UCE — 不可纠正错误
- PE — 奇偶校验错误

不要担心他们是否注意到一些错误，因为启动时可能会发生这种情况。运行时字段不应递增。如果是，则表明交换矩阵中存在问题。发出以下命令，以便了解每个交换矩阵平面的错误细目：

```
admin show controllers fabric plane <0-7> statistics detail
```

控制平面概述

线卡机箱和交换矩阵机箱之间的控制平面连接当前通过RP(LCC)和SCGE(FCC)上的千兆以太网端口进行。端口之间的互连通过一对Catalyst 6500交换机提供，该对交换机可通过两个或多个千兆以太网端口连接。



138147

Catalyst 6500配置

建议对用于多机箱控制平面的Catalyst交换机进行以下配置：

- 所有端口上都使用一个VLAN。
- 所有端口都以接入模式运行（无中继）。
- 生成树802.1w/s用于环路预防。
- 两台或多台链路用于交叉连接两台交换机，STP用于防止环路。不建议使用信道。
- 连接到CRS-1 RP和SCGE的端口使用预标准模式，因为IOS-XR不支持基于标准的802.1s。
- 应在交换机之间以及交换机与RP/SCGE之间连接的端口上启用UDLD。
- CRS-1上默认启用UDLD。

有关如何在[多机架系统中配置Catalyst 6500的详细信息](#)，请参阅在多机架系统上启动Cisco IOS XR软件。

多机箱控制平面管理

为多机箱系统提供控制平面连接的Catalyst 6504-E机箱为以下管理服务配置：

- 通过端口千兆1/2进行带内管理，该端口连接到每个PoP处的LAN交换机。仅允许小范围的子网和协议访问。
- NTP用于设置系统时间。
- 对标准主机执行系统日志记录。
- 可以为关键功能启用SNMP轮询和陷阱。

注意：在运行中不对Catalyst进行任何更改。应对任何计划的更改进行预先测试，强烈建议在维护期间执行此操作。

以下是管理配置示例：

```
#In-band management connectivity
interface GigabitEthernet2/1
  description *CRS Multi-chassis Management Ethernet - DO NOT TOUCH*
  ip address [ip address] [netmask]
  ip access-group control_only in
!
!
ip access-list extended control_only
  permit udp [ip address] [netmask] any eq snmp
  permit udp [ip address] [netmask] eq ntp any
  permit tcp [ip address] [netmask] any eq telnet

#NTP

ntp update-calendar
ntp server [ip address]

#Syslog
logging source-interface Loopback0
logging [ip address]
logging buffered 4096000 debugging
no logging console

#RADIUS
aaa new-model
aaa authentication login default radius enable
enable password {password}
radius-server host [ip address] auth-port 1645 acct-port 1646
radius-server key {key}

#Telnet and console access
!
access-list 3 permit [ip address]
!
line con 0
  exec-timeout 30 0
  password {password}
line vty 0 4
  access-class 3 in
  exec-timeout 0 0
  password {password}
```

[ROMMON和Monlib](#)

Cisco monlib是一个可执行程序，存储在设备上，并加载到RAM中，供ROMMON执行。ROMMON使用monlib来访问设备上的文件。ROMMON版本可以升级，应在思科技术支持的建议下

进行升级。最新的ROMMON版本为1.40。

升级说明

请完成以下步骤：

1. 从Cisco CRS-1 ROMMON下载[ROMMON二进制文件\(仅限注册客户\)](#)。
2. 解压缩TAR文件，并将6个BIN文件复制到Disk0的CRS根目录。

```
RP/0/RP0/Router#dir disk0:/*.bin
```

```
Directory of disk0:
```

```
65920      -rwx  360464      Fri Oct 28 12:58:02 2005  rommon-hfr-ppc7450-sc-dsmp-A.bin
66112      -rwx  360464      Fri Oct 28 12:58:03 2005  rommon-hfr-ppc7450-sc-dsmp-B.bin
66240      -rwx  376848      Fri Oct 28 12:58:05 2005  rommon-hfr-ppc7455-asm-p-A.bin
66368      -rwx  376848      Fri Oct 28 12:58:06 2005  rommon-hfr-ppc7455-asm-p-B.bin
66976      -rwx  253904      Fri Oct 28 12:58:08 2005  rommon-hfr-ppc8255-sp-A.bin
67104      -rwx  253492      Fri Oct 28 12:58:08 2005  rommon-hfr-ppc8255-sp-B.bin
```

3. 使用show diag | inc ROM|NODE|PLIM命令以查看当前rommon版本。

```
RP/0/RP0/CPU0:ROUTER(admin)#show diag | inc ROM|NODE|PLIM
```

```
NODE 0/0/SP : MSC(SP)
  ROMMON: Version 1.19b(20050216:033352) [CRS-1 ROMMON]
PLIM 0/0/CPU0 : 40C192-POS/DPT
  ROMMON: Version 1.19b(20050216:033559) [CRS-1 ROMMON]
NODE 0/2/SP : MSC(SP)
  ROMMON: Version 1.19b(20050216:033352) [CRS-1 ROMMON]
PLIM 0/2/CPU0 : 8-10GbE
  ROMMON: Version 1.19b(20050216:033559) [CRS-1 ROMMON]
NODE 0/4/SP : Unknown Card Type
NODE 0/6/SP : MSC(SP)
  ROMMON: Version 1.19b(20050216:033352) [CRS-1 ROMMON]
PLIM 0/6/CPU0 : 160C48-POS/DPT
  ROMMON: Version 1.19b(20050216:033559) [CRS-1 ROMMON]
NODE 0/RP0/CPU0 : RP
  ROMMON: Version 1.19b(20050216:033559) [CRS-1 ROMMON]
NODE 0/RP1/CPU0 : RP
  ROMMON: Version 1.19b(20050216:033559) [CRS-1 ROMMON]
NODE 0/SM0/SP : FC/S
  ROMMON: Version 1.19b(20050216:033352) [CRS-1 ROMMON]
NODE 0/SM1/SP : FC/S
  ROMMON: Version 1.19b(20050216:033352) [CRS-1 ROMMON]
NODE 0/SM2/SP : FC/S
  ROMMON: Version 1.19b(20050216:033352) [CRS-1 ROMMON]
NODE 0/SM3/SP : FC/S
  ROMMON: Version 1.19b(20050216:033352) [CRS-1 ROMMON]
```

4. 进入ADMIN模式，使用upgrade rommon a all disk0命令升级ROMMON。

```
RP/0/RP0/CPU0:ROUTER#admin
```

```
RP/0/RP0/CPU0:ROUTER(admin)#upgrade rommon a all disk0
```

```
Please do not power cycle, reload the router or reset any nodes until
all upgrades are completed.
```

```
Please check the syslog to make sure that all nodes are upgraded successfully.
```

```
If you need to perform multiple upgrades, please wait for current upgrade
to be completed before proceeding to another upgrade.
```

```
Failure to do so may render the cards under upgrade to be unusable.
```

5. 退出管理模式并输入show log |包含“OK，ROMMON A”，并确保所有节点都成功升级。如果任何节点发生故障，请返回步骤4并重新编程。

```
RP/0/RP0/CPU0:ROUTER#show logging | inc "OK, ROMMON A"
```

```
RP/0/RP0/CPU0:Oct 28 14:40:57.223 PST8: upgrade_daemon[380][360]: OK, ROMMON A is
programmed successfully. SP/0/0/SP:Oct 28 14:40:58.249 PST8: upgrade_daemon[125][121]: OK,
ROMMON A is programmed successfully. SP/0/2/SP:Oct 28 14:40:58.251 PST8:
```

```
upgrade_daemon[125][121]: OK, ROMMON A is programmed successfully. LC/0/6/CPU0:Oct 28
14:40:58.336 PST8: upgrade_daemon[244][233]: OK, ROMMON A is programmed successfully.
LC/0/2/CPU0:Oct 28 14:40:58.365 PST8: upgrade_daemon[244][233]: OK, ROMMON A is programmed
successfully. SP/0/SM0/SP:Oct 28 14:40:58.439 PST8: upgrade_daemon[125][121]: OK, ROMMON A
is programmed successfully. SP/0/SM1/SP:Oct 28 14:40:58.524 PST8: upgrade_daemon[125][121]:
OK, ROMMON A is programmed successfully. LC/0/0/CPU0:Oct 28 14:40:58.530 PST8:
upgrade_daemon[244][233]: OK, ROMMON A is programmed successfully. RP/0/RP1/CPU0:Oct 28
14:40:58.593 PST8: upgrade_daemon[380][360]: OK, ROMMON A is programmed successfully.
SP/0/6/SP:Oct 28 14:40:58.822 PST8: upgrade_daemon[125][121]: OK, ROMMON A is programmed
successfully. SP/0/SM2/SP:Oct 28 14:40:58.890 PST8: upgrade_daemon[125][121]: OK, ROMMON A
is programmed successfully. SP/0/SM3/SP:Oct 28 14:40:59.519 PST8: upgrade_daemon[125][121]:
OK, ROMMON A is programmed successfully.
```

6. 进入ADMIN模式，然后使用upgrade rommon b all disk0 命令升级ROMMON。

```
RP/0/RP0/CPU0:ROUTER#admin
RP/0/RP0/CPU0:ROUTER(admin)#upgrade rommon b all disk0
Please do not power cycle, reload the router or reset any nodes until
all upgrades are completed.
Please check the syslog to make sure that all nodes are upgraded successfully.
If you need to perform multiple upgrades, please wait for current upgrade
to be completed before proceeding to another upgrade.
Failure to do so may render the cards under upgrade to be unusable.
```

7. 退出管理模式并输入show log |包含“OK，ROMMON B”，并确保所有节点都成功升级。如果任何节点发生故障，请返回步骤4并重新编程。

```
RP/0/RP0/CPU0:Router#show logging | inc "OK, ROMMON B"
RP/0/RP0/CPU0:Oct 28 13:27:00.783 PST8: upgrade_daemon[380][360]: OK,
ROMMON B is programmed successfully.
LC/0/6/CPU0:Oct 28 13:27:01.720 PST8: upgrade_daemon[244][233]: OK,
ROMMON B is programmed successfully.
SP/0/2/SP:Oct 28 13:27:01.755 PST8: upgrade_daemon[125][121]: OK,
ROMMON B is programmed successfully.
LC/0/2/CPU0:Oct 28 13:27:01.775 PST8: upgrade_daemon[244][233]: OK,
ROMMON B is programmed successfully.
SP/0/0/SP:Oct 28 13:27:01.792 PST8: upgrade_daemon[125][121]: OK,
ROMMON B is programmed successfully.
SP/0/SM0/SP:Oct 28 13:27:01.955 PST8: upgrade_daemon[125][121]: OK,
ROMMON B is programmed successfully.
LC/0/0/CPU0:Oct 28 13:27:01.975 PST8: upgrade_daemon[244][233]: OK,
ROMMON B is programmed successfully.
SP/0/6/SP:Oct 28 13:27:01.989 PST8: upgrade_daemon[125][121]: OK,
ROMMON B is programmed successfully.
SP/0/SM1/SP:Oct 28 13:27:02.087 PST8: upgrade_daemon[125][121]: OK,
ROMMON B is programmed successfully.
RP/0/RP1/CPU0:Oct 28 13:27:02.106 PST8: upgrade_daemon[380][360]: OK,
ROMMON B is programmed successfully.
SP/0/SM3/SP:Oct 28 13:27:02.695 PST8: upgrade_daemon[125][121]: OK,
ROMMON B is programmed successfully.
SP/0/SM2/SP:Oct 28 13:27:02.821 PST8: upgrade_daemon[125][121]: OK,
ROMMON B is programmed successfully.
```

8. upgrade 命令只会使用新的ROMMON烧录bootflash的特殊保留部分。但新的ROMMON在重新加载卡之前保持非活动状态。因此，当您重新加载卡时，新的ROMMON处于活动状态。每次重置一个节点或重置整个路由器以执行此操作。

```
Reload Router:
RP/0/RP0/CPU0:ROUTER#hw-module node 0/RP0/CPU0 or 0/RP1/CPU0 reload (depends on which on is
in Standby Mode.
RP/0/RP0/CPU0:ROUTER#reload
!--- Issue right after the first command. Updating Commit Database. Please wait...[OK]
Proceed with reload? [confirm] !--- Reload each Node. For Fan Controllers (FCx), !--- Alarm
Modules (AMx), Fabric Cards (SMx), and RPs (RPx), !--- you must wait until the reloaded
node is fully reloaded !--- before you reset the next node of the pair. But non-pairs !---
can be reloaded without waiting. RP/0/RP0/CPU0:ROUTER#hw-module node 0/RP0/CPU0 or
0/RP1/CPU0 reload
!--- This depends on which on is in Standby Mode. RP/0/RP0/CPU0:ROUTER#hw-module node
```

0/FC0/SP

RP/0/RP0/CPU0:ROUTER#hw-module node 0/AM0/SP

RP/0/RP0/CPU0:ROUTER#hw-module node 0/SM0/SP

!--- Do not reset the MSC and Fabric Cards at the same time. RP/0/RP0/CPU0:ROUTER#hw-module node 0/0/CPU

9. 使用show diag | inc ROM|NODE|PLIM命令以检查当前ROMMON版本。

```
RP/0/RP1/CPU0:CRS-B(admin)#show diag | inc ROM|NODE|PLIM
```

```
NODE 0/0/SP : MSC(SP)
```

```
ROMMON: Version 1.32(20050525:193402) [CRS-1 ROMMON]
```

```
PLIM 0/0/CPU0 : 40C192-POS/DPT
```

```
ROMMON: Version 1.32(20050525:193559) [CRS-1 ROMMON]
```

```
NODE 0/2/SP : MSC(SP)
```

```
ROMMON: Version 1.32(20050525:193402) [CRS-1 ROMMON]
```

```
PLIM 0/2/CPU0 : 8-10GbE
```

```
ROMMON: Version 1.32(20050525:193559) [CRS-1 ROMMON]
```

```
NODE 0/6/SP : MSC(SP)
```

```
ROMMON: Version 1.32(20050525:193402) [CRS-1 ROMMON]
```

```
PLIM 0/6/CPU0 : 160C48-POS/DPT
```

```
ROMMON: Version 1.32(20050525:193559) [CRS-1 ROMMON]
```

```
NODE 0/RP0/CPU0 : RP
```

```
ROMMON: Version 1.32(20050525:193559) [CRS-1 ROMMON]
```

```
NODE 0/RP1/CPU0 : RP
```

```
ROMMON: Version 1.32(20050525:193559) [CRS-1 ROMMON]
```

```
NODE 0/SM0/SP : FC/S
```

```
ROMMON: Version 1.32(20050525:193402) [CRS-1 ROMMON]
```

```
NODE 0/SM1/SP : FC/S
```

```
ROMMON: Version 1.32(20050525:193402) [CRS-1 ROMMON]
```

```
NODE 0/SM2/SP : FC/S
```

```
ROMMON: Version 1.32(20050525:193402) [CRS-1 ROMMON]
```

```
NODE 0/SM3/SP : FC/S
```

```
ROMMON: Version 1.32(20050525:193402) [CRS-1 ROMMON]
```

注意：在CRS-8和交换矩阵机箱上，ROMMON还将风扇速度设置为默认速度4000 RPM。

PLIM和MSC概述

这表示CRS-1路由器上的数据包流，这些术语可互换使用：

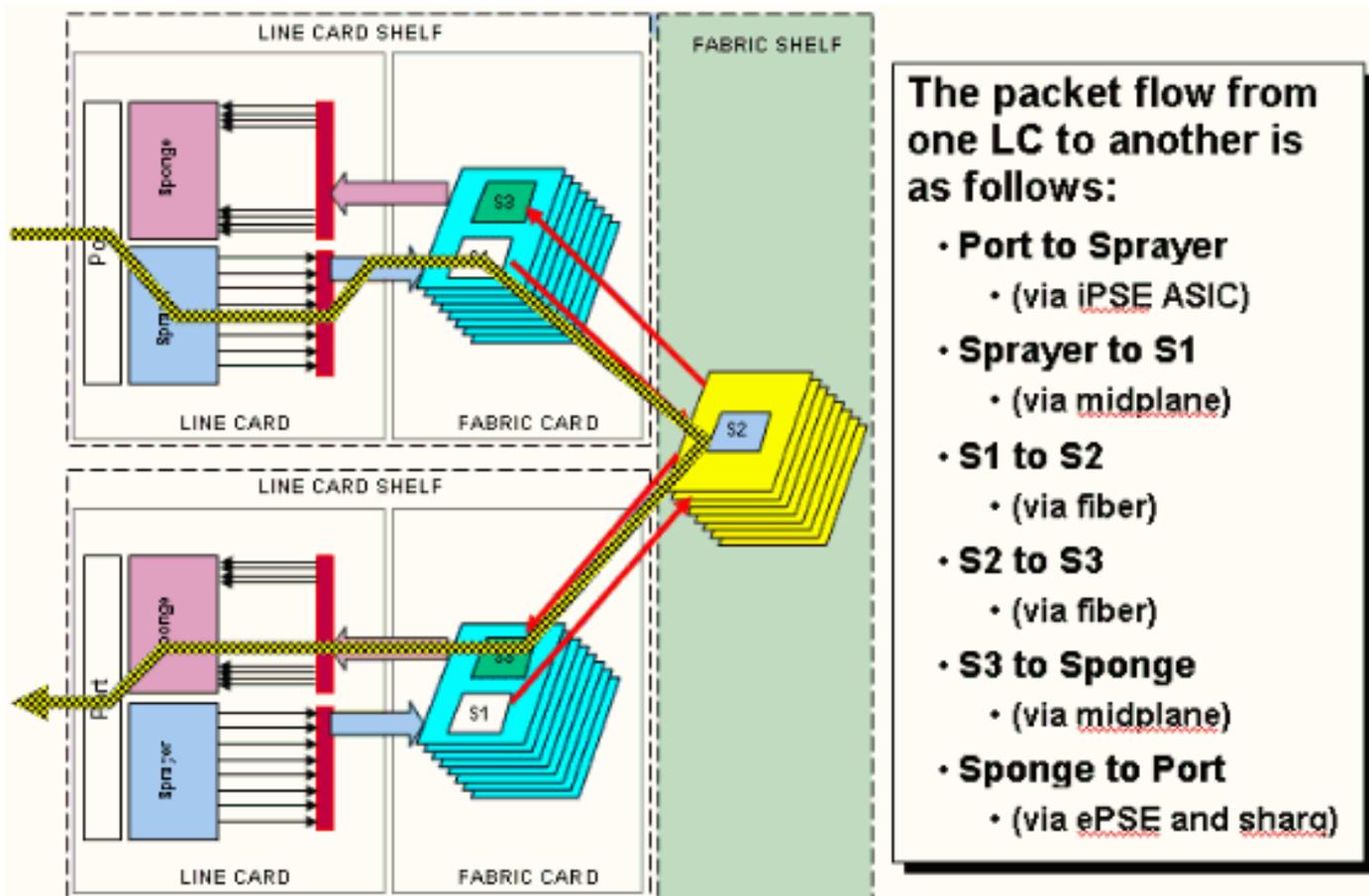
IngressQ ASIC也称为喷雾器ASIC。

FabricQ ASIC也称为海绵ASIC。

EgressQ ASIC也称为Sharq ASIC。

SPP也称为PSE (数据包交换引擎) ASIC。

Rx PLIM > Rx SPP > Ingress Q > Fabric > Fabric Q > Tx SPP > Egress Q > Tx PLIM (喷雾器) (海绵) (沙克)

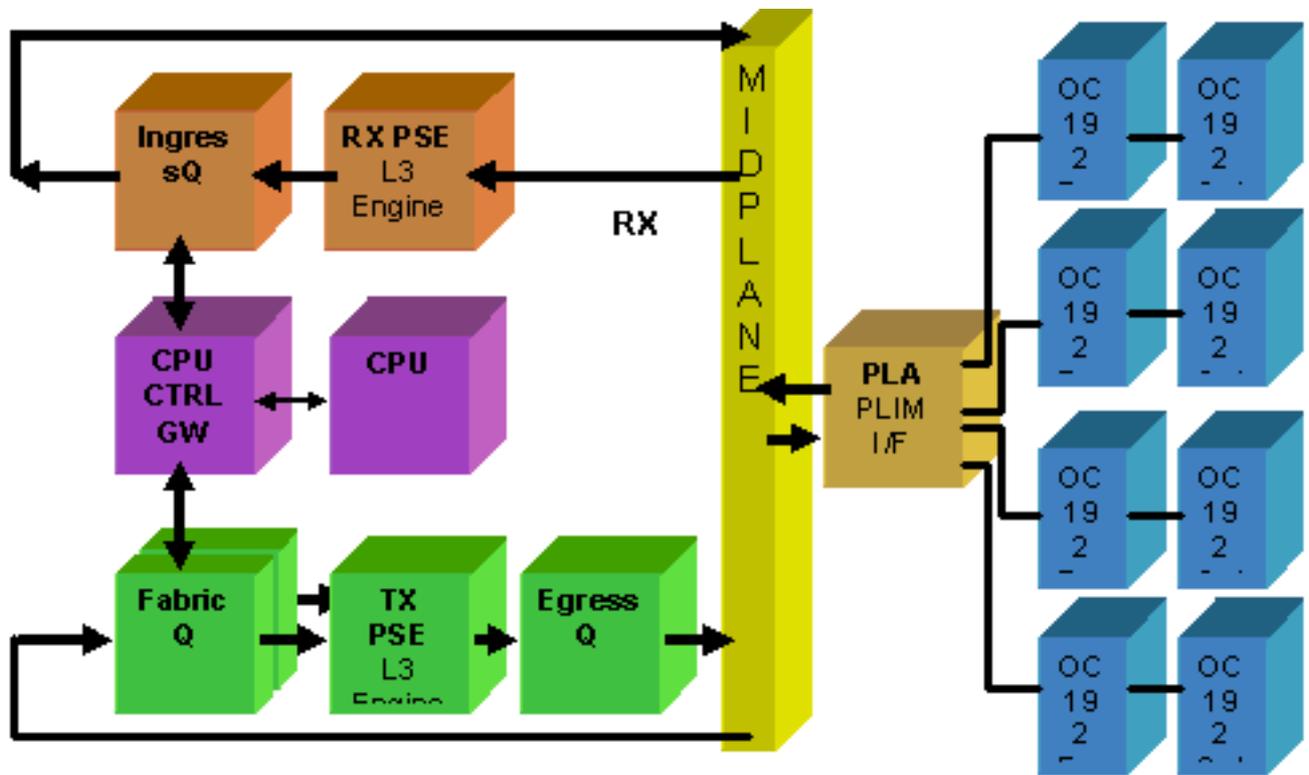


数据包在物理层接口模块(PLIM)上接收。

PLIM包含与其相配的MSC的物理接口。PLIM和MSC是通过机箱背板连接的独立卡。因此，特定MSC的接口类型由与其匹配的PLIM的类型定义。根据PLIM的类型，卡包含各种ASIC，为接口提供物理介质和成帧。PLIM ASIC的目的是提供MSC和物理连接之间的接口。它端接光纤，进行光到电转换，终止介质成帧，即SDH/Sonet/Ethernet/HDLC/PPP，检查CRC，添加一些控制信息，称为缓冲区报头，并将保留的位转发到MSC。PLIM不对HDLC或PPP保持连接进行源/接收。这些由MSC上的CPU处理。

PLIM还提供以下功能：

- 1/10千兆以太网的MAC过滤
- 1/10千兆以太网的入口/出口MAC记帐
- 1/10千兆以太网的VLAN过滤
- 1/10千兆以太网的VLAN记帐
- 入口缓冲和拥塞通知



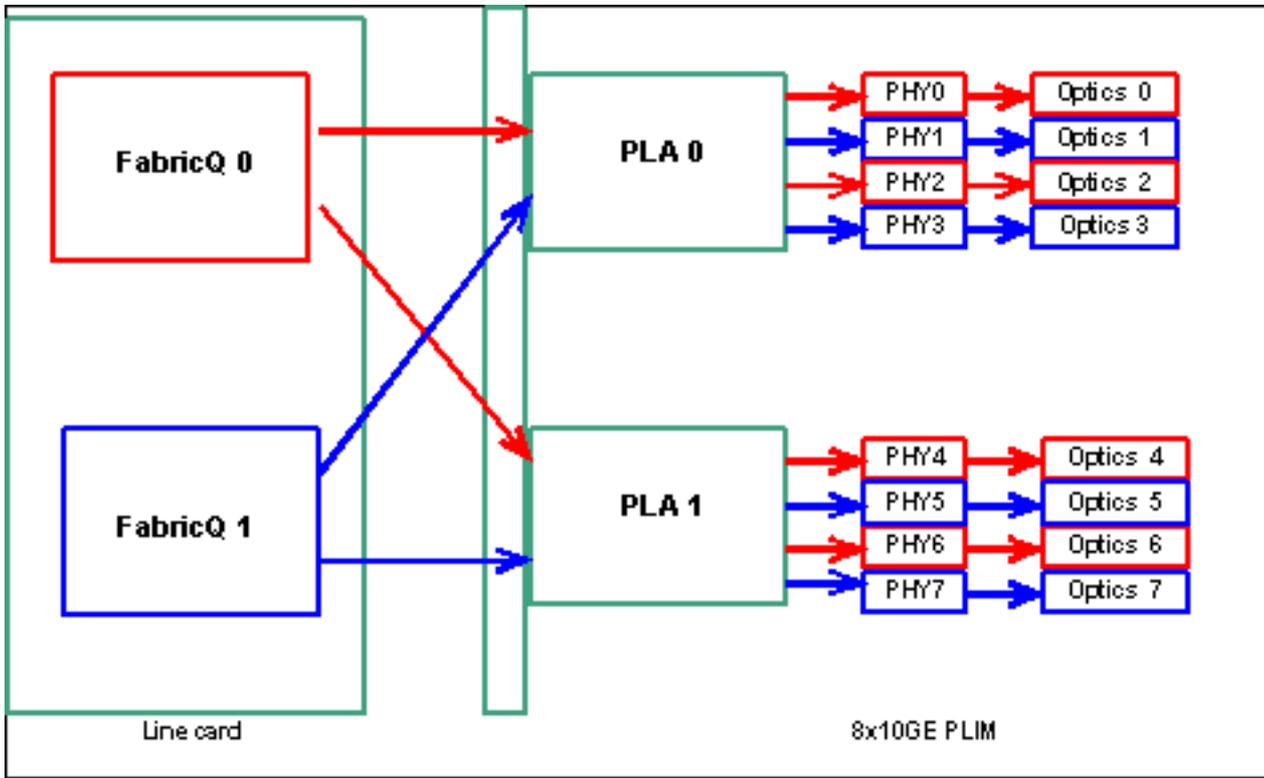
PLIM超订用

10GE PLIM

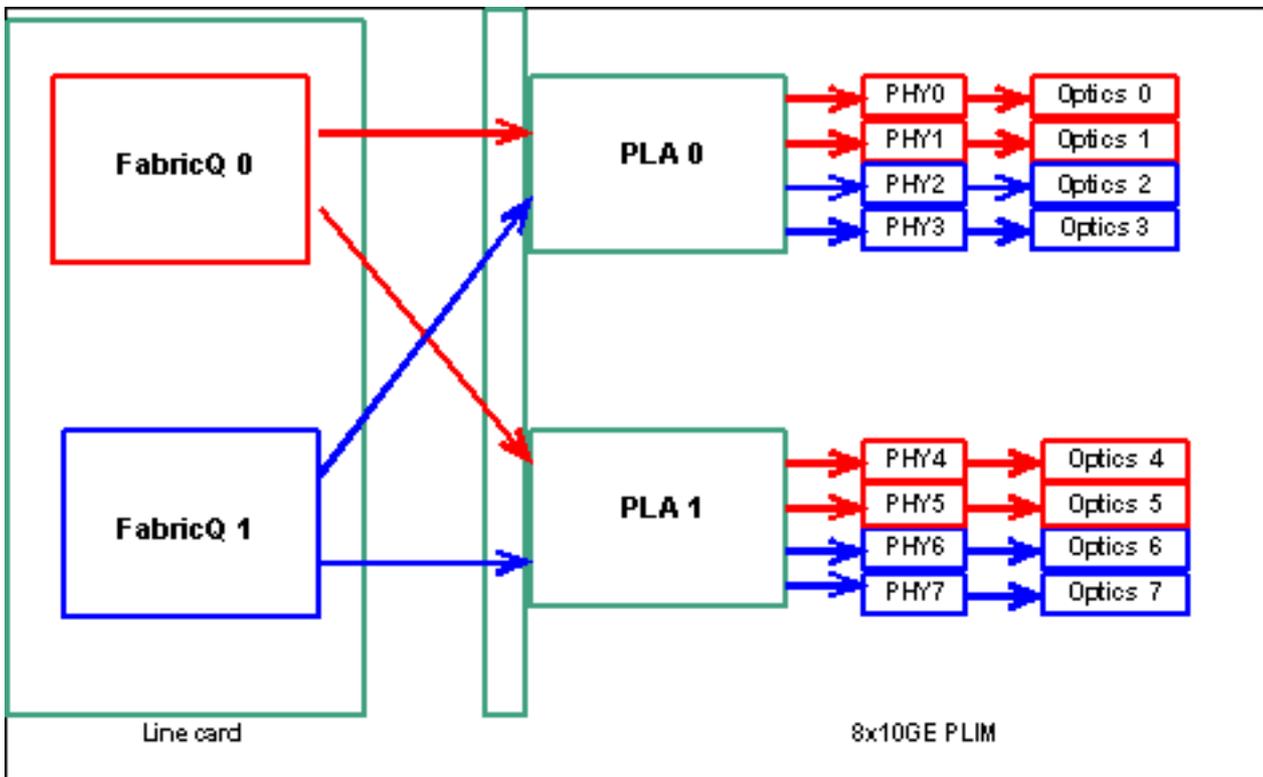
8 X 10G PLIM能够终止约80 Gbps的流量，而MSC的转发容量最大为40 Gbps。如果PLIM上所有可用端口都已填充，则会发生超订用，QoS建模将变得极其重要，以确保不会无意中丢弃高级流量。对一些人而言，超额认购不是一种选择，必须避免。为此，八个端口中只有四个必须使用。此外，必须小心确保MSC和PLIM内的最佳带宽可用于四个端口中的每个端口。

注意：从版本3.2.2开始，端口映射会更改。请参见这些图。

端口映射到版本3.2.1



从版本3.2.2开始的端口映射



如前所述，物理端口由两个FabricQ ASIC之一提供服务。端口到ASIC的分配是静态定义的，不能更改。此外，8 X 10G PLIM有两个PLA ASIC。第一个PLA服务端口0到3，第二个服务4到7。在8 X 10G PLIM上，单个PLA的带宽容量约为24 Gbps。单个FabricQ ASIC的交换容量约为62 Mpps。

如果填充端口0到3或端口4到7，则PLA(24 Gbps)的带宽容量在所有四个端口之间共享，这些端口限制了整体吞吐量。如果填充端口0、2、4和6 (最多3.2.1) 或0、1、4和5 (从3.2.2开始)，因为所有这些端口都由一个FabricQ ASIC提供服务，该ASIC的交换容量同样为62 Mpps，这会限制吞吐量容量。

最好以能获得PLA和FabricQ ASIC最高效率的方式利用端口，以实现最佳性能。

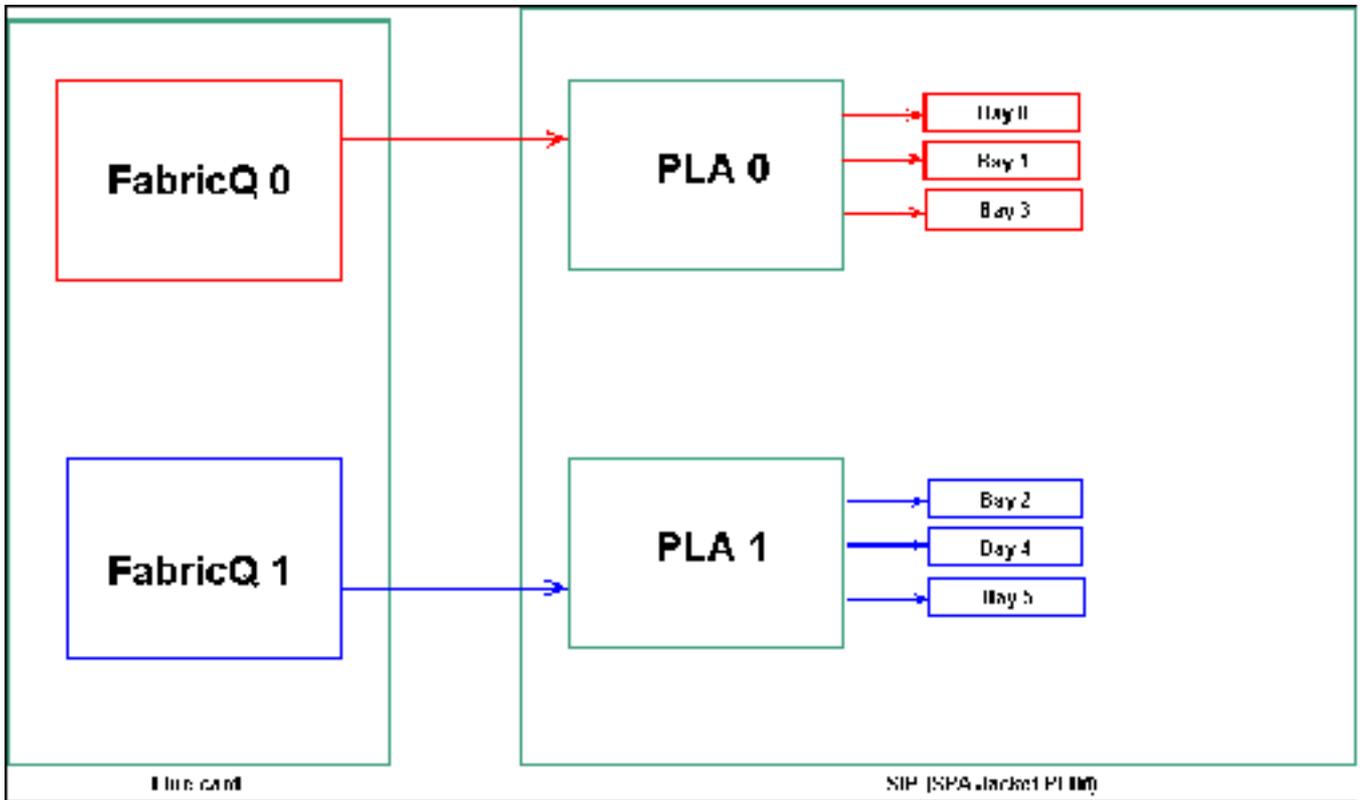
[SIP-800/SPA](#)

SIP-800 PLIM能够使用称为服务端口适配器(SPA)的模块化接口卡运行。SIP-800提供6个SPA托架，理论接口容量为60 Gbps。MSC的转发容量最大为40 Gbps。如果要填充SIP-800上的所有托架，则根据SPA类型，可能会发生超订用，QoS建模变得极其重要，以确保不会无意中丢弃高级流量。

注意：POS接口不支持超订用。但是，必须适当放置10 Gb POS SPA，以确保提供正确的吞吐量容量。10 Gb以太网SPA仅在IOS-XR版本3.4中受支持。此SPA提供超订用功能。

对一些人而言，超额订购不是一种选择，必须避免。为此，必须只使用六个托架中的四个。此外，必须小心确保MSC和PLIM内的最佳带宽可用于四个端口中的每个端口。

[SPA湾映射](#)



如前所述，物理端口由两个FabricQ ASIC之一提供服务。端口到ASIC的分配是静态定义的，不能更改。此外，SIP-800 PLIM有两个PLA ASIC。第一个人民解放军服务端口0、1和3，第二个服务端口2、4和5。

SIP-800 PLIM上单个PLA的带宽容量约为24 Gbps。单个FabricQ ASIC的交换容量约为62 Mpps。

如果填充端口0、1和3或端口2、4和5，则PLA(24 Gbps)的带宽容量将在所有三个端口之间共享，这些端口会限制整体吞吐量。由于每个FabricQ都为这些端口组提供服务，因此端口组的最大数据包速率为62 Mpps。最好以获得PLA最高效率的方式利用端口，以实现最佳带宽。

建议的位置：

	SPA湾#	SPA湾#	SPA湾#	SPA湾#
第1项	0	1	4	5
第2项	1	2	3	4

如果要用四个以上的SPA填充卡，建议填写以前列出的选项之一，该选项会在两个端口组（0、1和3 & 2、4和5）之间分布接口。然后，您应将下一个SPA模块放在0、1和3以及2、4和5端口组中的一个开放端口中。

DWDM XENPACK

从版本3.2.2开始，DWDM XENPACK可以安装并提供可调光纤模块。此类XENPACK模块的冷却要求要求在安装的模块之间有一个空白插槽。此外，如果安装了单个DWDM XENPACK模块，则最多可使用四个端口，即使XENPACK模块不是DWDM设备也是如此。因此，这对FabricQ到PLA到端口的映射有直接影响。需要注意此要求，并在下表中加以考虑。

建议的位置：

	光纤端口号	光纤端口号	光纤端口号	光纤端口号
选项1或DWDM XENPACK	0	2	5	7
第2项	1	3	4	6

对于3.2.2或更高版本或3.3安装，请避免FabricQ映射更改。因此，更简单的放置模式可用于常规和DWDM XENPACK模块。

	光纤端口号	光纤端口号	光纤端口号	光纤端口号
第1项	0	2	4	6
第2项	1	3	5	7

如果要用四个以上的非DWDM XENPACK端口填充卡，建议完成列出的选项之一，该选项将光纤接口模块分布在两个端口组（0-3和4-7）之间。然后，您需要将下一个光纤接口模块放在0-3或4-7端口组中的一个开放端口中。如果将0-3端口组用于光接口模块#5，则光接口模块#6应放在4-7端口组中。

有关详细信息，请参阅DWDM XENPAK模块。

配置管理

IOS-XR中的配置通过两阶段配置完成，配置由用户在第一阶段输入。这是CLI仅检查配置语法的阶段。此阶段中输入的配置仅为配置代理进程（例如CLI/XML）所知。未验证配置，因为它未写入sysdb服务器。后端应用程序不会收到通知，并且无法访问或了解此阶段的配置。

在第二阶段，配置由用户明确提交。在此阶段，配置将写入sysdb服务器，后端应用程序验证配置和通知是由sysdb生成的。在提交在第一阶段中输入的配置之前，可以中止配置会话。因此，不安全地假设在阶段1中输入的所有配置始终在阶段2中提交。

此外，路由器的操作和/或运行配置可由多个用户在第1阶段和第2阶段期间修改。因此，在第1阶段运行配置和/或运行状态的路由器的任何测试在实际提交配置的第2阶段可能无效。

配置文件系统

配置文件系统(CFS)是用于存储路由器配置的一组文件和目录。CFS存储在RP上使用的默认介质disk0:/config/目录下。CFS中的文件和目录是路由器内部的，用户不应修改或删除。这可能导致配置丢失或损坏并影响服务。

每次提交后，CFS都会检查到备用RP。这有助于在故障切换后保留路由器的配置文件。

在路由器启动期间，从CFS中存储的配置提交数据库应用最后的活动配置。用户无需在每次配置提交后手动保存活动配置，因为这是由路由器自动完成的。

在启动期间应用配置时，不建议进行配置更改。如果配置应用程序未完成，则登录路由器时会看到以下消息：

系统配置过程

此设备的启动配置当前正在加载。这可能需要几分钟。完成后会通知您。在此过程完成之前，请勿尝试重新配置设备。在某些情况下，最好从用户提供的ASCII配置文件恢复路由器配置，而不是从CFS恢复最后一个活动配置。

您可以通过以下方式强制应用配置文件：

```
using the "-a" option with the boot command. This option forces
the use of the specified file only for this boot.
```

```
rommon>boot <image> -a <config-file-path>
```

```
setting the value of "IOX_CONFIG_FILE" boot variable to the
path of configuration file. This forces the use of the specified file
for all boots while this variable is set.
```

```
rommon>IOX_CONFIG_FILE=
```

```
rommon>boot <image>
```

恢复路由器配置时，一个或多个配置项可能无法生效。所有失败的配置都保存在CFS中，并在下次重新加载之前一直保留。

您可以浏览失败的配置、解决错误并重新应用配置。

以下是一些提示，用于解决路由器启动期间配置失败的问题。

在IOX中，配置可归类为失败配置，原因有三：

1. 语法错误 — 解析器生成语法错误，通常表示与CLI命令不兼容。您应更正语法错误并重新应用配置。
2. 语义错误 — 当配置管理器在路由器启动期间恢复配置时，后端组件会生成语义错误。请注意，cfgmgr不负责确保配置作为运行配置的一部分被接受。Cfgmgr只是中间人，只报告后端组件生成的任何语义故障。由每个后端组件所有者分析故障原因并确定故障原因。用户可以在配置模式下执行describe <CLI命令>，以便轻松找到后端组件验证器的所有者。例如，如果路由器bgp 217显示为失败配置，则describe命令显示组件验证器是ipv4-bgp组件。

```
RP/0/0/CPU0:router#configure terminal
```

```
RP/0/0/CPU0:router(config)#describe router bgp 217
The command is defined in bgpv4_cmds.parser
```

```
Node 0/0/CPU0 has file bgpv4_cmds.parser for boot package /gsr-os-mbi-3.3.87/mbi12000-rp.vm
from gsr-rout
```

```
Package:
```

```
  gsr-rout
```

```
    gsr-rout V3.3.87[Default] Routing Package
```

```
    Vendor  : Cisco Systems
```

```
    Desc    : Routing Package
```

```
    Build   : Built on Mon Apr  3 16:17:28 UTC 2006
```

```
    Source  : By ena-view3 in /vws/vpr/mletchwo/cfgmgr_33_bugfix for c2.95.3-p8
```

```
    Card(s) : RP, DRP, DRPSC
```

```
    Restart information:
```

```
      Default:
```

```
        parallel impacted processes restart
```

```
Component:
```

```
  ipv4-bgp V[fwd-33/66] IPv4 Border Gateway Protocol (BGP)
```

```
File: bgpv4_cmds.parser
```

```
User needs ALL of the following taskids:
```

```
  bgp (READ WRITE)
```

```
It will take the following actions:
```

```
  Create/Set the configuration item:
```

```
    Path: gl/ip-bgp/0xd9/gbl/edm/ord_a/running
```

```
    Value: 0x1
```

```
Enter the submode:
```

```
  bgp
```

```
RP/0/0/CPU0:router(config)#
```

3. 应用错误 — 配置已成功验证并作为运行配置的一部分被接受，但后端组件由于某种原因无法更新其运行状态。由于配置已正确验证，因此在两个运行配置中都显示该配置，并且由于后端操作错误而显示为失败的配置。**describe**命令可以再次在未能应用的CLI上运行，以查找组件应用所有者。要在启动操作符期间浏览和重新应用失败的配置，请完成以下步骤：对于R3.2操作员，可使用以下步骤重新应用失败的配置：操作员可以使用**show configuration failed startup**命令浏览路由器启动期间保存的失败配置。操作员应运行**show configuration failed startup noerror | file myfailed.cfg**命令，以便将启动失败的配置保存到文件。操作员应进入配置模式并使用**load/commit**命令以重新应用此失败配置：

```
RP/0/0/CPU0:router(config)#load myfailed.cfg
```

```
Loading.
```

```
197 bytes parsed in 1 sec (191)bytes/sec
```

```
RP/0/0/CPU0:router(config)#commit
```

对于R3.3映像，操作员可以使用以下更新过程：操作员必须使用**show configuration failed startup**命令和**load configuration failed startup**命令来浏览和重新应用任何失败的配置。

```
RP/0/0/CPU0:router#show configuration failed startup
```

```
!! CONFIGURATION FAILED DUE TO SYNTAX/AUTHORIZATION ERRORS
```

```
telnet vrf default ipv4
```

```
server max-servers 5 interface POS0/7/0/3 router static
```

```
address-family ipv4 unicast
```

```
 0.0.0.0/0 172.18.189.1
```

```
!! CONFIGURATION FAILED DUE TO SEMANTIC ERRORS
```

```
router bgp 217 !!%
```

```
Process did not respond to sysmgr !
```

```
RP/0/0/CPU0:router#
```

```

RP/0/0/CPU0:router(config)#load configuration failed startup noerror
Loading.
263 bytes parsed in 1 sec (259)bytes/sec
RP/0/0/CPU0:mike3(config-bgp)#show configuration
Building configuration...
telnet vrf default ipv4 server max-servers 5 router static
address-family ipv4 unicast
  0.0.0.0/0 172.18.189.1
  !
!
router bgp 217
!
end

RP/0/0/CPU0:router(config-bgp)#commit

```

内核转储程序

默认情况下，如果进程崩溃，IOS-XR会将核心转储写入硬盘，但如果内核本身崩溃，则不会。请注意，对于多机箱系统，目前仅线卡机箱0支持此功能。其他机箱在未来的软件版本中支持。

建议在标准和管理模式配置中使用以下配置来启用RP和MSC的内核转储：

```

exception kernel memory kernel filepath harddisk:
exception dump-tftp-route port 0 host-address 10.0.2.1/16 destination 10.0.2.1 next-hop 10.0.2.1
tftp-srvr-addr 10.0.2.1

```

内核转储配置

这会导致内核崩溃发生以下情况：

1. RP崩溃，转储会写入该RP上位于磁盘根目录的硬盘。
2. 如果MSC崩溃，转储将写入磁盘根目录中RP0的硬盘。

这对RP故障切换时间没有影响，因为为路由协议配置了不间断转发(NSF)。崩溃的RP或线路卡在写入核心时发生崩溃后可能需要额外几分钟才能再次可用。

此处显示了将此配置添加到标准和管理模式配置的示例。请注意，管理模式配置需要使用DRP。

此输出显示内核转储配置示例：

```

RP/0/RP0/CPU0:crs1#configure
RP/0/RP0/CPU0:crs1(config)#exception kernel memory kernel filepat$
RP/0/RP0/CPU0:crs1(config)#exception dump-tftp-route port 0 host-$
RP/0/RP0/CPU0:crs1(config)#commit
RP/0/RP0/CPU0:crs1(config)#
RP/0/RP0/CPU0:crs1#admin
RP/0/RP0/CPU0:crs1(admin)#configure
Session                Line                User                Date                Lock
00000201-000bb0db-00000000 snmp                hfr-owne           Wed Apr  5 10:14:44 2006
RP/0/RP0/CPU0:crs1(admin-config)#exception kernel memory kernel f$
RP/0/RP0/CPU0:crs1(admin-config)#exception dump-tftp-route port 0$
RP/0/RP0/CPU0:crs1(admin-config)#commit
RP/0/RP0/CPU0:crs1(admin-config)#
RP/0/RP0/CPU0:crs1(admin)#

```



```
Success rate is 97 percent (97/100), round-trip min/avg/max = 1/2/5 ms
RP/0/RP0/CPU0:ss01-crs-1_P1#show lpts pifib hardware entry statistics location 0/5/CPU0 | excl
0/0
```

```
* - Vital; L4 - Layer4 Protocol; Intf - Interface;
DestAddr - Destination Fabric Address;
na - Not Applicable or Not Available
```

Local, Remote Address.Port	L4	Intf	DestAddr	Pkts/Drops
-----	-----	-----	-----	--- any
any any Punt 100/3				
224.0.0.5 any	any	PO0/5/1/0	0x3e	4/0
224.0.0.5 any	any	PO0/5/1/1	0x3e	4/0

<further output elided>

IPsec

IP数据包本身是不安全的。IPsec是用于保护IP数据包的方法。CRS-1 IPsec在软件转发路径中实施，因此IPsec会话在RP/DRP上终止。每个CRS-1支持的IPsec会话总数为500。数量取决于CPU速度和已分配的资源。此功能没有软件限制，只有RP上的本地源和本地终止的流量才符合IPsec处理的条件。IPsec传输模式或隧道模式均可用于流量类型，但前者是首选，因为IPsec处理的开销较少。

R3.3.0支持BGP和OSPFv3的IPsec加密。

有关如[何实施IPsec的详细信息](#)，请参阅《Cisco IOS XR系统安全配置指南》。

注意：IPsec需要加密饼，例如，hfr-k9sec-p.pie-3.3.1。

带外

控制台和AUX访问

CRS-1 RP/SC既有可用于带外管理的控制台端口和AUX端口，也有通过IP实现带外管理的以太网端口。

每个RP/SCGE的控制台和AUX端口（每个机箱两个）可以连接到控制台服务器。这意味着单机箱系统需要四个控制台端口，而多机箱系统需要12个端口和另外两个端口，用于Catalyst 6504-E上的Supervisor引擎。

AUX端口连接非常重要，因为它提供对IOS-XR内核的访问，并且当无法通过控制台端口进行恢复时，它允许系统恢复。通过AUX端口的访问仅对系统上本地定义的用户可用，并且仅当用户具有根系统或思科支持级别访问时才可用。此外，用户必须定义**加密**密码。

虚拟终端访问

Telnet和安全外壳(SSH)可用于通过vty端口到达CRS-1。默认情况下，两者均被禁用，用户需要明确启用它们。

注意：IPsec需要加密饼，例如，hfr-k9sec-p.pie-3.3.1。

首先生成RSA和DSA密钥，如本示例所示，以启用SSH:

```
RP/0/RP1/CPU0:Crs-1#crypto key zeroize dsa
```

% Found no keys in configuration.

RP/0/RP1/CPU0:CrS-1#**crypto key zeroize rsa**

% Found no keys in configuration.

RP/0/RP1/CPU0:CrS-1#**crypto key generate rsa general-keys**

The name for the keys will be: the_default

Choose the size of the key modulus in the range of 360 to 2048 for your General Purpose Keypair.

Choosing a key modulus greater than 512 may take a few minutes.

How many bits in the modulus [1024]:

Generating RSA keys ...

Done w/ crypto generate keypair

[OK]

RP/0/RP1/CPU0:CrS-1#**crypto key generate dsa**

The name for the keys will be: the_default

Choose the size of your DSA key modulus. Modulus size can be 512, 768, or 1024 bits. Choosing a key modulus

How many bits in the modulus [1024]:

Generating DSA keys ...

Done w/ crypto generate keypair

[OK]

!--- VTY access via SSH & telnet can be configured as shown here. vty-pool default 0 4 ssh
server ! line default secret cisco users group root-system users group cisco-support exec-
timeout 30 0 transport input telnet ssh ! ! telnet ipv4 server

[相关信息](#)

- [路由器支持](#)
- [技术支持和文档 - Cisco Systems](#)